3-D MODEL BASED APPROACH TO VIDEO COMPRESSION

by

P. ANANDH

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

APRIL, 1997

3-D MODEL BASED APPROACH TO VIDEO COMPRESSION

A Thesis Submitted

in Partial Fulfillment of the Requirements

for the Degree of

Master of Technology

by

P. Anandh

to the

DEPARTMENT OF ELECTRICAL ENGINEERING,
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

April 1997

CERTIFICATE

This is to certify that the work contained in this M. Tech thesis entitled **3-D MODEL BASED APPROACH TO VIDEO COMPRESSION** *has been carried out by* **P. Anandh** *under my supervision and has not been submitted elsewhere for a degree.*

Administrative Supervisor
Dr. Govind Sharma
Associate Professor
Dept. of Electrical Engg.
Indian Institute of Technology
Kanpur.

Supervisor
Dr. Sumana Gupta
Associate Professor
Dept. of Electrical Engg.
Indian Institute of Technology
Kanpur.

Dedicated to
my parents.

Acknowledgement

I would take this opportunity to express my sincere gratitude towards **Dr. Sumana Gupta**, my thesis supervisor for her invaluable guidance. She has always been very patient and encouraging regarding the doubts and discussions during the thesis work. I would like to thank my friends sar, gomes and sen who were helping me during the thesis. I would also like to thank ksri, muthup, bsbabu, vvk, ksub, tsb, swami, vrajesh, narasi, edhan, suyog, nime, intekhab who were the support of my iitk career.

I would like to thank Silicon Automation Systems(SAS) for their financial support during my fourth semester.

Abstract

In this thesis we have attempted an implementation of an object based analysis-synthesis coder (OBASC), low bitrate video coder for videophone application. The method uses the concepts of a model-based approach. In this method the parts of the given sequence of image frames are separated out. They are referred to as objects. Each object is defined by a set of three parameters namely shape, motion and color respectively. For each frame they are estimated and transmitted. In the decoder the image is reconstructed using these transmitted parameters. Unlike the conventional hybrid coder, it has been shown that the use of shape information avoids the mosquito and blocking artifacts in the reconstructed image. The number of bits needed to code the motion and shape parameter of an object have been found to be 50 and 70 bits respectively for each frames. Coding of texture has not been attempted. Assuming the texture of an object requires 1.2kb for each frame, it can be concluded that the overall bitrate required for a head and shoulder image sequence with a frame rate of 10Hz is approximately 64kb/sec.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

The digital representation of an image or of a sequence of images requires a very large number of bits. The goal of image compression is to reduce this number, as much as possible and to reconstruct a faithful duplicate of the original picture. International standardization efforts related to different applications have been made. Examples include, digital transmission of high density television (HDTV) aiming at bitrates of 20Megabits/sec(Mb/s), MPEG 2 (Moving picture expert group) for broadcast television at bitrates of 10Mb/s, MPEG 1 for digital video at bitrates of 2Mb/s and H.261 standard for low bitrate, 64Kb/s, videophone and video conferencing coding.

Efforts are continuously devoted towards new standards, one of them being MPEG 4 to be used for transmitting video signals at very low bitrates, below 64kb/s. Video coding at very low bitrates is motivated by its potential applications for *videophones, multimedia electronic mail, remote sensing, electronic newspapers, interactive multimedia databases, multimedia annotation, surveillance, telemedicine and communication aids for deaf people.*

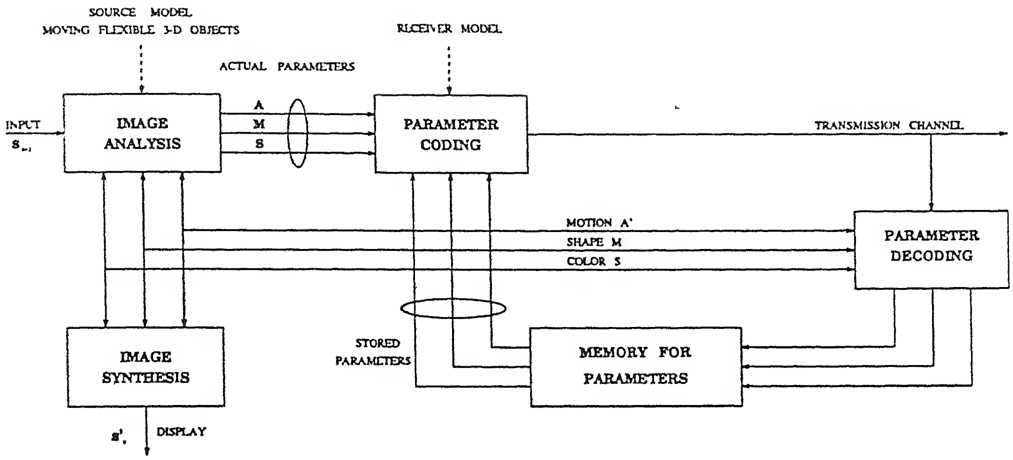

Figure 1.1: Block diagram of an object-based analysis- synthesis coder.

Due to practical medium capacity limitations, the main problem of introducing these applications lies in how to compress a huge amount of visual information into a very low bitrate stream for transmission/storage purpose. With this in view, we are primarily concerned with the design of a video coder based on the concept of model-based image coding for videophone applications.

The work presented in this thesis is concerned with the development of an object based analysis-synthesis coder (OBASC). The block diagram of OBASC coder that has been implemented is shown in fig 1.1. For each input frame a set of 3 parameters namely motion,shape and color are extracted and stored in the memory as shown in the figure. The stored parameters in the memory are fedback to the image analysis block to estimate the set of parameter of the current frame. The stored parameters are also used to encode the estimated parameter of the current frame. The coded parameters are transmitted and also decoded in order to avoid the accumulation of coding and analysis errors. A detailed description of the method is given in the subsequent chapters.

1.1 Image Compression Methods for Videophone Image Sequences

The *videotelephone* problem can be defined as the problem of compressing the huge amount of visual information into a very low bitrate stream for transmission. It is the problem [2] of transmitting the videotelephone scenes through the available public switched telephone network (PSTN). It is known that the available PSTN network is mainly used for transmission of speech. Visual data is considerably larger than the speech data. If we adopt the CIF (common intermediate format) standard for a digital picture format needed for a low bitrate video, the bitrate of the CIF sequence is approximately 37 Mb/s. On trying to transmit a color video signal via the PSTN under the assumption that channel capacity is extended to 16 kb/s, and using 8 kb/s for video and voice, respectively,the compression ratio needed is higher than 4500. Achieving such a high compression ratio indeed poses a serious challenge to the researchers in the image coding field.

It is impossible to compress a full TV signal at very high compression ratios while still keeping high quality of the decoded images when transmitted via the PSTN. There are certain restrictions which are implied in videotelephone application. As reported in [2], typical videophone scenes have the following three special characteristics:

1. **Fixed scene content** The typical scene is a head and shoulder image of the speaker. Due to the objects of the scene being known *a priori* , some knowledge about them can be used.
2. **Limited motion** The interfering motion is mainly caused by the movement of the speaker and the camera is generally fixed. This situation is not valid for a mobile videotelephone, but even in this case, the camera

undergoes limited motion, such as zoom, pan and vibration. The movement of the speaker mainly contains the global movement of the head and shoulder and the local motion of facial expression changes. Due to the inertia of the human body, the global motion is relatively slow and can be described using only few bits per frame. In this way, more bits can be spent on facial expressions.

3. **Special requirements for visual information** Interpersonal video communications does not usually require the full resolution that is provided by broadcast television. The key in visual communication is to provide the emotional dimensions. Therefore, a lower resolution image format is often used.

One commonly used format is the QCIF in which the resolution is reduced to 144 × 180 for luminance and 72 × 90 for chrominance. The frame rate is reduced to 10Hz or even 6Hz. The combined content of the knowledge of the scene, the spatio-temporal redundancy etc. allows the visual information to be compressed to obtain very high compression ratio. A brief review [2] of the techniques for very low bitrate image sequence coding is discussed in this section.

R.Wallis proposed an ultra-low bandwidth video conferencing system which could be operated at 9.6kb/s bitrate. The compression is achieved by transforming the original grey level image into a binary image. 2-D run-length coding techniques are then used to compress the binary images. Lippman proposed two approaches to transmit a videophone scene. In the first approach referred to as storage based coding, the images to be displayed are known in advance and therefore transmitted at full resolution and with full dynamics by local retrieval. In the second scheme only the instructions, to retrieve those images or sequences are transmitted.

The coding schemes given above are aimed at 8kb/s, and have low resolution with low frame rate. For videoconferencing, these drawbacks are overcome by allowing a transmission bitrate of 64kb/s. Methods used to achieve this can be divided into two categories: *waveform-based coding* and *model-based coding* techniques respectively.

In waveform-based coding an image is treated as a 2-D signal waveform or a segment of an image sequence as a 3-D signal waveform exploiting the inherent statistical or deterministic properties. Most image coding techniques such as transform coding, subband/wavelet coding, VQ coding and fractal coding can be included in this group. These methods fail at the boundaries of the naturally moving objects and causes coding artifacts known as blocking and mosquito effects [7]. In order to avoid these coding distortions, an alternative method object based analysis-synthesis coding (OBASC) technique is introduced to obtain bitrates of the order of 64kb/s.

1.2 Thesis Overview

The thesis work is organized in the following manner. In chapter 2, the model-based image coding methods are discussed. The typical scene specific knowledge needed in videophony are formulated and presented. In chapter 3, the OBASC method implemented in this thesis is explained. The concepts of model world and model objects are introduced. The initialization procedure used in the coder followed by a detailed discussion of the image analysis and synthesis is included in this chapter. Chapter 4 is concerned with the estimation of motion, shape and color parameters of an object. Chapter 5 presents the results and concludes the work undertaken. The scope for future work is also included chapter 5.

Chapter 2

MODEL BASED CODING

In model-based image coding the input image is viewed as a 2-D projection of a 3-D real world (scene). The coding is performed by first modeling the 3-D scene, extracting the model parameters at the encoder and finally synthesizing the image at the decoder by using the extracted and quantized parameters.

If we can reconstruct the three dimensional scene model that leads to 2-D image sequence, and the images are analyzed and synthesized based on this model, then a great reduction in image information can be expected. This is the basic idea of the *model-based coding* method. The basic schematic of a model-based coder is shown in figure 2.1. Three key elements in model-based coding are *the modeling*, that is, the reconstruction of the 3-D scene model, *image analysis* and *image synthesis* based on the scene model.

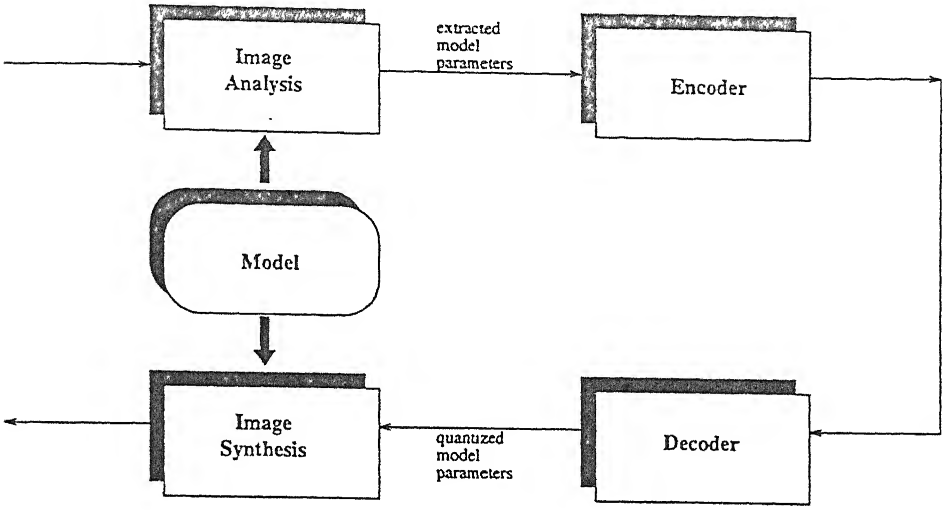

Figure 2.1: Basic schematic diagram of model-based image coding

2.1 Exploiting Some Scene Specific Knowledge

The task of estimating 3D parameters from a 2D image is difficult even for simple image. The algorithms can be simplified using the knowledge of the scenes of a videotelephone. To completely exploit the scene specific knowledge, a strategy has been developed for video telephone. This has been reported as a set of hypothesis [6].

Hypothesis 1:

The image consists of a moving person in front of constant background. The changes between two frames are only due to the movement of an object. It governs the initial model generation. An intial 3-D object shape is modeled, based on the change detection silhouette [5].

Hypothesis 2:

The object exhibits rigid 3D motion. It is assumed that the object consists of one rigid component and the mean 3-D motion is estimated. This global motion compensation reduces the mean estimation error.

Hypothesis 3:

The object may consists a set of flexibly connected rigid components. The residual error is analyzed in order to search for clusters of rigid motion components. The object is split into a connected set of rigid components and 3-D motion of each object is estimated.

Hypothesis 4:

It is assumed that the true 3-D shape of the person is unknown but time invariant and only the facial expression changes with time. This allows attribution of fast temporal changes to flexible texture deformations and the long term shape deviation is interpreted as an incorrect 3-D shape. The shape changes are recorded over time, the mean shape is adapted and fast flexible distortions are corrected from frame to frame.

Hypothesis 5:

The texture of an object is time invariant. Some small regions (eyes and mouth) can be time variant. The remaining intensity errors between synthesized and the real image sequences are attributed to changes in the surface texture and failure of model assumptions. Those areas that exhibit annoying visual distortions in the image are detected as model failures areas and surface texture from the real image is inserted into the model through texture mapping

These hypothesis will simplify the algorithm of OBASC method when used for videotelephone application.

2.2 Image Modeling

Modeling normally consists of two parts: structure modeling and motion modeling. According to the representation of structure and motion information, the models can be roughly split into two groups: explicit models and implicit models. In the former case, 3-D structure and motion of scenes are explicitly modeled while in the latter they are taken into account implicitly. Employing an explicit 3-D model, 2-D images can be more accurately described and more easily manipulated.

Example for explicitly modeled coding is *sementic-based image coding.* In this coding [2], the model is explicitly defined such as, the human head, to analyze and synthesize the moving images.

The implicit model includes *object-oriented image coding*, in which no explicit model is used. This scheme can be applied to a more general class of objects. In addition, a recognition process is not required in this type of coding method. Hence it has a lower complexity. If an appropriate model is chosen in the implicit method, then its performance in terms of its coding efficiency and subjective quality of image reconstruction will be the same as that of explicit method.

With a 3-D model. hidden surfaces or occlusions can be removed from

Examples	Surface-Based model	Volume-Based model
Parametric model	Spline Harmonic surface	Generalized Cylinder Superquadrics
Nonparametric model	Wireframe	Voxels

Table 2.1: Classification of scene models

2-D projected images, which is very hard to achieve by using only 2-D models. When the explicit 3-D structure of scenes are not accessible or are difficult to describe by using an explicit model, an implicit model may be favourable.

In this section some image models that are potentially applicable to model-based coding and the related modeling techniques are discussed.

2.2.1 Image models

Since the motion description is often implicitly related to the structure description used, we mainly examine geometric models for structure description. Geometric models can be classified into two classes: (a) *volume based description* and (b) *surface based description* It can be further classified as either a parametric or a nonparametric model. Table 2.1 reflects the interaction of these two kinds of classifications.

In OBASC method the surface based description is used. The major advantage of scene representation based on the surface primitives is that such descriptions are easily converted into surface representation which favours image rendering. In surface based description, *nonparametric wireframe models* are most popular. In these models the surface is approximated by planar polygonal patches, generally triangular patches. In a triangular wireframe model.

the surface shape is represented by the set of points defining the vertices of these triangles. Since the size of the patches is adjustable according to the surface complexity, wireframe representation is flexible. Therefore, wireframe models are extensively used for model-based coding [2].

Most world objects are solids, although usually only their surfaces are visible. Therefore a description based on volumetric primitives is a natural approach to model objects. Most existing investigations on volume based models focus on the *parametric volume description.* An often used set of volumetric primitives is the class of *generalized cylinders (GC)* . GC was used on early work in biological form analysis for fitting real data. Using the deformable cylinder model it is able to generate 3-D models of some natural objects from 2-D silhouette. Superquadrics is another alternative to generalized cylinders [11][12]. Superquadrics and its variation with dynamic global and local deformations encompass a large variety of natural shapes. This type of primitive is capable of modeling nonrigid motion because any nonrigid deformation can be achieved by pushing, pinching and pulling on a lump of elastic material. A comparison between the surface based representation and the volume based representation is shown in Table 2.2.

Table 2.2: Comparison between surface graphics and volume graphics.

Sl.	Capability	Surface Graphics	Volume Graphics
1.	Rendering performance	Sensitive to scene and object complexity	Insensitive to scene and object complexity
2.	Memory and processing requirement	Variable: depends on scene and object complexity	Large but constant
3.	Object-space aliasing	None	Frequent
4.	Transformation	Continuous; Performed on the geometric definitions of objects	Discrete; Performed on voxel subvolumes
5.	Scan conversion and rendering	Pixelization and embedded in viewing	voxelization is decoupled from viewing
6.	Boolean and block operations	Difficult; must be performed analytically	Trivial: by using voxbit, voxel-by-voxel operation, aggregation, octrees

Table continued on next page.

Sl.	Capability	Surface Graphics	Volume Graphics
7.	Rendering of interior and amorphous phenomena	No; surfaces only	Yes; rendering of inner structures as well as surfaces
8.	Adequacy for sampled data and intermixing with geometric data	Partially and indirectly(fitting followed by surface rendering)	supports a representation and direct rendering
9.	Measurements(for example, distance, area, volume, normal)	Analytical, but may be complex	Discrete approximation, but simple
10.	Viewpoint dependency	Requires recalculation for every viewpoint change	Precomputes and stores view point independent information

Chapter 3

BASIC CONCEPTS OF OBJECT BASED ANALYSIS-SYNTHESIS CODING

3.1 Object Oriented Image Coding

Object oriented image coding was first proposed by Musmann and consequently improved upon by Hotter [5] and Ostermann [7]. The basic idea behind this coding scheme is to reconstruct a model world which has a model image identical to the real image. A world is described by a scene, its illumination and its camera. The model world consists of objects described by a set of parameters. Object based analysis-synthesis coder (OBASC) subdivides an image into 'm' different objects. Each object is defined by three sets of parameters $A^{(m)}, M^{(m)}$ and $S^{(m)}$ describing its motion, shape and color respectively. Motion parameters define the motion of the object, shape parameters give the shape and position of object in model world and color parameter gives the luminance and chrominance reflectance of the surface of the object. These are also referred to as textures. The block diagram of an OBASC coder is shown in the fig 1.1. Instead of frame memory used in conventional video coders, OBASC requires memory for storing the transmitted object parameter. The parameters of an object are encoded and transmitted for each frames. The parameter memories in the coder and decoder contain the same information.

The task of image analysis block is to analyze the current image and estimate the parameters of each object using the knowledge of the decoded parameters of the preceding frame s_k'. The transmitted parameters are stored in the buffer, The feedback of the coded parameters to the analysis system is needed to avoid the accumulation of coding and analysis errors.

The first frame is compressed using the conventional coding method and transmitted. For successive frames, only the coded parameters are transmitted. The color information for the object in the successive frames is taken from the first frame. In the current image moving objects are detected. New motion $A_{k+1}^{(m)}$ and shape $M_{k+1}^{(m)}$ parameters for these objects are estimated and the color information of the previous frame $S_k^{(m)}$ can be used for the present frame. Those objects for which both motion and shape are estimated successfully are called as model compliance(MC-object). These objects are defined by the parameter sets $A_{k+1}^{(m)}, M_{k+1}^{(m)}, S_k^{'(m)}$. The parameter $S_k^{'(m)}$ is the quantized color parameter of the previous frame. Finally, the image areas which cannot be described by MC-objects, using new shape, motion and transmitted color parameters are detected. These areas are referred to as model failures. Some authors label it as special objects.

After the 3D-motion for MC-objects is estimated and compensated, the remaining area will contain small position and shape errors of MC-objects-referred to as geometrical distortions. These will not affect the subjective quality of the image. Thus model failures are restricted to those image areas with significant differences between the motion and shape compensated predicted image s_k^* and current image s_{k+1} giving annoying visual distortions. Their size is very small, of the order of 4% of the image area.

3.2 Source Model

The source model used here assumes a 3-D real world which has to be modeled by a 3-D model world. While the real image is taken by a real camera looking into the real world, a model image is synthesized using a model camera looking into the model world. A world is described by a scene, its illumination and its camera. A scene consisting of objects, their motion and their relative position. The image area representing the projection of an object is referred to as projection m.

The goal of the modeling is to generate a model world W_k with a model image identical to the real image s_k at the time instant k. This implies that the model objects may differ from the real objects. An example is shown in the fig 3.1, where the real and model objects are not identical, but their projections are identical. However similarity between real objects and model objects generally helps performing proper image analysis.

The real illumination is modeled by constant diffuse illumination and the real camera by a pinhole camera whose target is the model image. The camera model is shown in fig 3.2. It is seen that the camera projects the point $P^{(i)} = (P_x^{(i)}, P_y^{(i)}, P_z^{(i)})^T$ of the scene onto the point $p^{(i)} = (p_X^{(i)}, p_Y^{(i)})^T$ of the image plane according to the following equation

$$p_X^{(i)} = F\frac{P_x^{(i)}}{P_z^{(i)}}$$

$$p_Y^{(i)} = F\frac{P_y^{(i)}}{P_z^{(i)}}$$

where F is the focal length of the camera, (X, Y) are image coordinates and (x, y, z) are model world coordinates as shown in fig 3.2.

The object source model taken into consideration is a rigid 3D object. The 3D motion can be represented by the rotation and translation vectors of the object for each frame. In the model used here, the shape parameters $M^{(m)}$

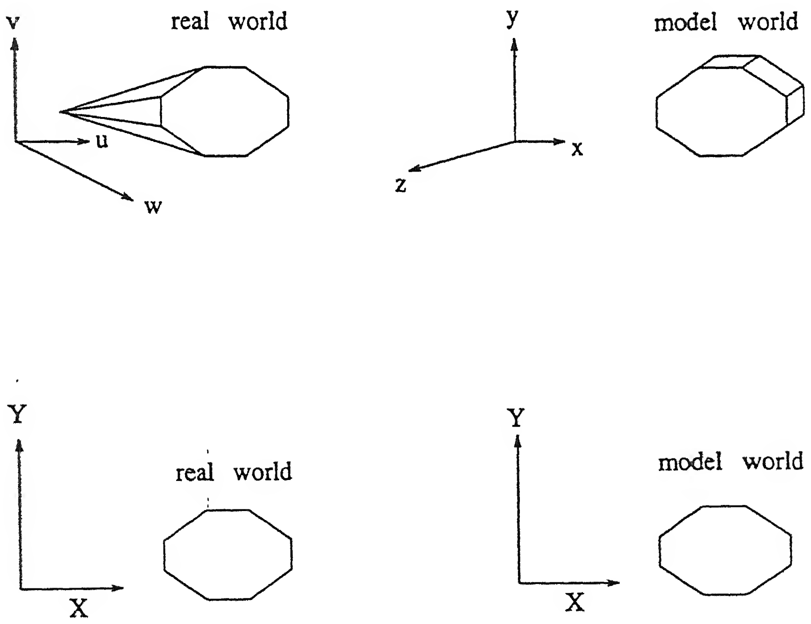

Figure 3.1: A real world and a model world where real and model images are identical.

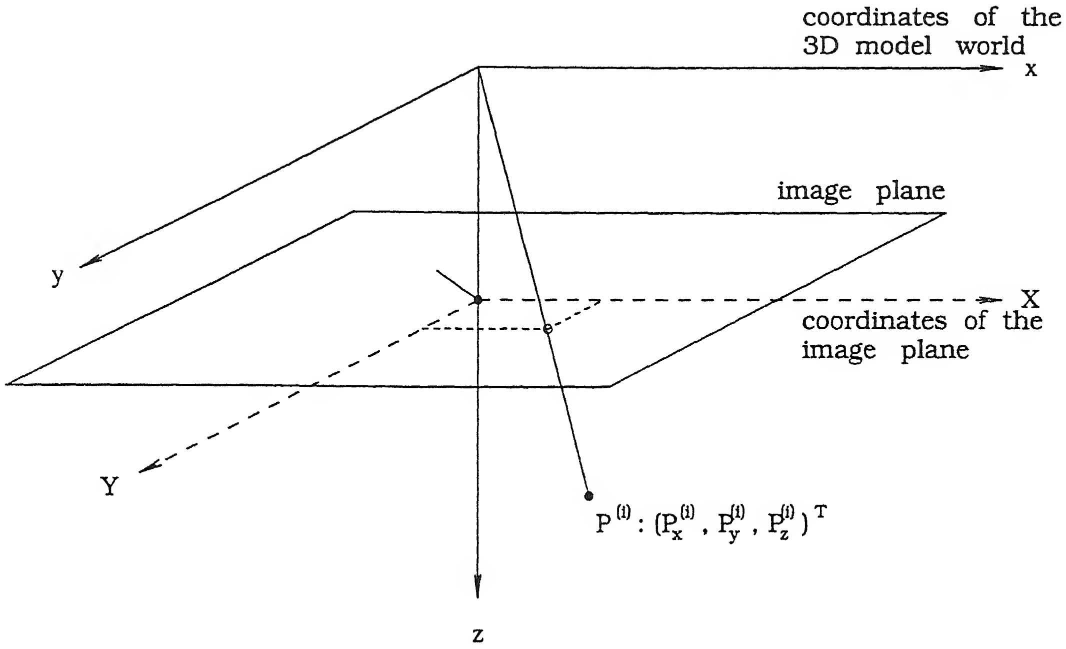
coordinates of the 3D model world
x
image plane
y
X
coordinates of the image plane
Y
$P^{(i)}: (P_x^{(i)}, P_y^{(i)}, P_z^{(i)})^T$
z

Figure 3.2: Camera model.

$M^{(m)}$ of an object m represent a 2-D binary mask which defines the *silhouette* of the model object in the model image. During initialization, the 3-D shape of the object is completely described by its 2-D silhouette. The algorithm that computes a 3-D shape from a 2-D silhouette will be discussed in Sec(3.3). After initialization, the shape parameters $M^{(m)}$ are used as update parameters to the model object shape. The 3-D shape is represented by a mesh of triangles which is put up by vertices referred as *control points* $\mathbf{P}_C^{(i)}$. The appearance of the model object surface is described by the color parameters $S^{(m)}$, which define the luminance and chrominance reflectance. An object may consist of two or more rigid components. Each component has its own set of motion parameters. Since each component is defined by its control points, the components are linked by those triangles of the object having control points belonging to different components. Due to these triangles, components are flexibly connected. A set of observation points is taken and 2D motion for each point is estimated. Using the information, 3D motion is estimated for each object. The shape for 3D rigid body need not be updated if the shape is defined precisely. Although the computation of shape information is easier for rigid body, 3-D flexible objects are more useful for representing natural objects like human face with different facial expressions.

The 3-D motion of the model objects are described by the parameters $A^{(m)} = (T_x^{(m)}, T_y^{(m)}, T_z^{(m)}. R_x^{(m)}, R_y^{(m)}, R_z^{(m)})$ defining translation and rotation of the object in x, y and z directions. A point $\mathbf{P}^{(i)}$ on the surface of object m with N control points $\mathbf{P}_C^{(i)}$ is moved to its new position $\mathbf{P}'^{(i)}$ according to

$$\mathbf{P}'^{(i)} = [\mathbf{R}_C^{(m)}](\mathbf{P}^{(i)} - \mathbf{C}^{(m)}) + \mathbf{C}^{(m)} + \mathbf{T}^{(m)} \tag{3.1}$$

with the translation vector $\mathbf{T}^{(m)}$,the object center $\mathbf{C}^{(m)}$ and the rotation matrix $[\mathbf{R}_C]$ defining the rotation in the mathematically positive direction around the x-,y- and z-axis with the rotation center $\mathbf{C}^{(m)}$.

We have $\mathbf{T}^{(m)} = (T_x^{(m)}, T_y^{(m)}, T_z^{(m)})^T$,

$\mathbf{C}^{(m)} = (C_x, C_y. C_z) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{P}_C^{(i)}$

and

$$[\mathrm{R}] = \begin{bmatrix} \cos R_y \cos R_z & \sin R_x \sin R_y \cos R_z - \cos R_x \sin R_z & \cos R_x \sin R_y \cos R_z + \sin R_x \sin R_z \\ \cos R_y \sin R_z & \sin R_x \sin R_y \sin R_z + \cos R_x \cos R_z & \cos R_x \sin R_y \sin R_z - \sin R_x \cos R_z \\ -\sin R_y & \sin R_x \cos R_y & \cos R_x \cos R_y \end{bmatrix}$$

From eqn(3.1) it is observed that $\mathbf{R}_C^{(m)}$ and $\mathbf{T}^{(m)}$ are independent of camera position.

3.3 Generation of the Model Object

The 3-D shape of an object can be generated from 2D silhouette of the object. In order to generate a 3D shape from an object silhouette a distance transformation is applied to the silhouette. Ellipse is used as a generating function giving the z-coordinate of the shape for each point of the silhouette as a function of its distance to the boundary of the silhouette. This generating function may be different for specific scenes. The ellipse as a generating function is a good initial guess for video telephone [7]. The use of generating function for computing the z-coordinates allows one to generate a model object from an arbitarily shaped silhouette. The parameter defining the major axis of the ellipse is the object width in the image plane. The maximum object depth is perpendicular to the image plane. For videophone sequences, subjectively good results are achieved with the ratio of objectwidth to object depth set to 1.5.

Once the 3-D shape is generated from its 2-D silhouette, contours on the model object were drawnsuch that the distance between two contour lines along the surface of the object is kept constant. The contour curves are on the surface of the model object. These curves are approximated by polygons. For this approximation of the contour lines by ploygons, the same error criterion d^*_{max} which will later be used for 2-D shape coding is applied. The quality measure d^*_{max} defines the maximum allowable distance between a contour line and its approximated polygon. The qualiity measure d^*_{max} in this case is taken to be 1.4 pels. The resulting polygon points are control points of the mesh of triangles. These mesh of triangles are generated after control points are determined. All the vertices of the triangles lie on the surface of the model object.

The coordinates of the points are generated from a distance transformation. They are not the exact coordinate lie on the surface of the object. These coordinates are to be updated for every frame to give a good estimate of 3D shape. The original image s_1 is now projected into the model scene using the geometory of the model camera. Motion and shape parameters are estimated for the model objects.

3.4 Image Synthesis

In OBASC, image synthesis is required for displaying the decoded picture at the receiver and for image analysis. For MC-objects it is enough to transmit the shape and motion parameters. The color information can be obtained from previous image by texture mapping [6]. Different procedures to generate a first geometric representation of the model scene were presented above. It is now necessary to determine the photometric properties of the observed objects. All photometric properties are summarized in a surface intensity map that is projected onto the camera target. For each triangular surface patch of the model object, the appropriate image patch is mapped onto the object surface in a texture map. When more than one view of the object is available, The view with the highest possible image resolution for each patch is selected as the texture map fig 3.3 explains the the process of texture mapping and image synthesis. The surface is approximated through plane triangular surface patches with control points P_1, P_2 and P_3. The control points are projected into image frame 1, and the circumscribing rectangle is stored as texture map for each triangle separately, which can be replaced throughout the sequence if needed. Through object or camera motion, the projection of the control points change from frame 1 to frame k. From these changes, an affine transform [6] that maps the image texture from the texture map onto image frame k is calculated. A point p_i in frame 1 is mapped to position p_i' in frame k, through the mapping equation

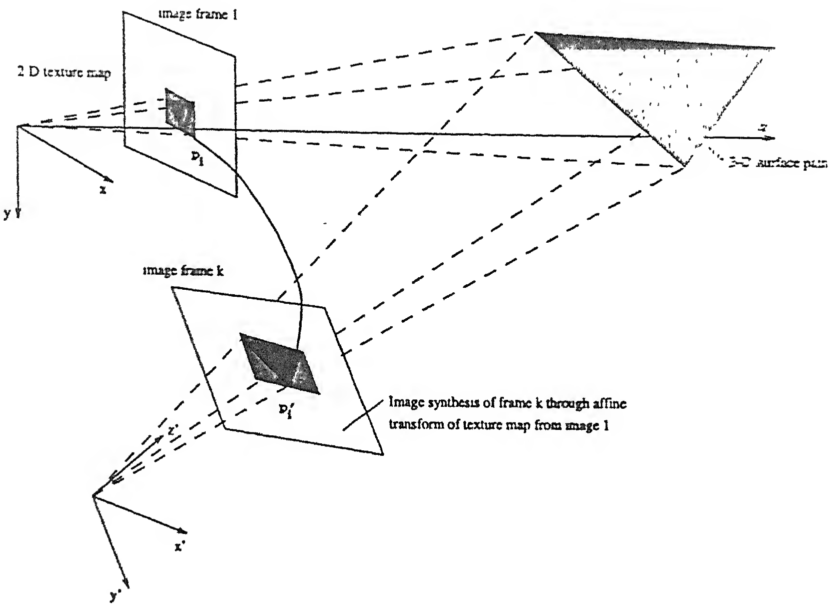
image frame 1
2 D texture map
p_i
x
y
z
3-D surface patch
image frame k
p_i'
Image synthesis of frame k through affine
transform of texture map from image 1
z'
x'
y'

Figure 3.3: Surface texure mapping and image synthesis.

$$\mathbf{p}_i' = \begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \end{pmatrix} \cdot \mathbf{p}_i + \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \tag{3.2}$$

The six parameters of the mapping equation (3.2) can be computed by inserting the perspective projection of control point P_i at frame 1 (p_i) and its projection at frame k as p_i' in Eqn(3.2) for each of the three control points P_1, P_2 and P_3 respectively. The resulting six linear equations are then solved for the parameters $a1, a2, b1, b2, c1$ and $c2$. Linear intensity interpolation is provided to map those intensity values where the transform projects a pixel between the regular pixel grid of the texture map. Mapping eqn (3.2) provides a transform that does not account for perspective foreshortening of the triangle, which will cause a geometric distortion. As long as a single triangle is small compared with its distance to the camera, this error can be neglected.

3.5 Image Analysis

The main aim of image analysis is to estimate the parameters $A^{(m)}_{k+1}, M^{(m)}_{k+1}, S^{(m)}_{k+1}$. and model failures taking the current image s_{k+1} and previousely transmitted parameters $A'^{(m)}_{k+1}, M'^{(m)}_{k+1}, S'^{(m)}_{k+1}$ to give a faithful reconstruction of the current image s_{k+1}. The overview of image analysis is shown in fig 3.4. From the prevousely transmitted parameters the previuous frame s'_k is synthesized. This image s'_k and the current image s_{k+1} are used for image analysis. From these images the 3D motion and shape are estimated for MC-objects and the image s^*_k is synthesized with these parameters. The synthesized image s^*_k and current image s_{k+1} are used to estimate the MF-objects and its parameters.

The figure 3.4 shows the structure of image analysis. Input to the image analysis are s_{k+1} and model world W'_k described by its parameters $A'^{(m)}_k$, $M'^{(m)}_k$, $S'^{(m)}_k$ for each object m. First, a model image s'_k of the current model world is computed by means of image synthesis.

In order to compute change detection mask B_{k+1}, the change detection evaluates the images s'_k and s_{k+1} based on hypothesis 1 (Sec 2.1). Fixing the background noise level (7/255 is used here) the change detection mask is computed. 2-D motion for observation points(to be discussed in Sec 4.2) are estimated and an averaging method is used to estimate the 3-D motion of MC-objects. In order to separate the moving objects from the uncovered background the motion is applied to the change detection mask B_{k+1} . The basic idea for the detection of uncovered background is that the projection of moving object before and after motion has to lie completely in the changed area [4]. Subtracting the uncovered background from the change detection mask B_{k+1} we obtain the new silhouette of the object. Thus shape adaptation is done after estimating the motion.

For detection of model failures, the prediction image is synthesized with

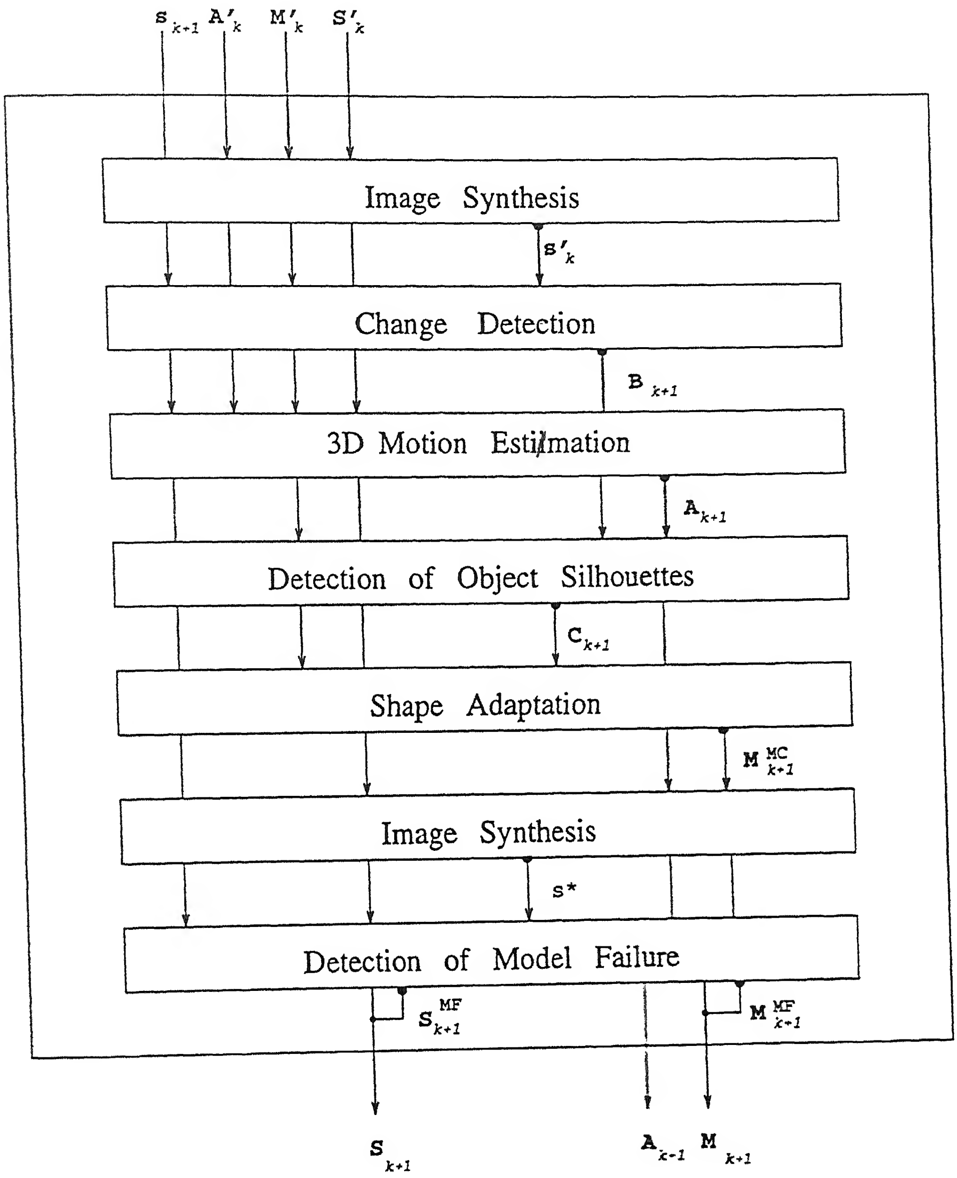

Figure 3.4: Block diagram of image analysis.

only shape and motion parameters of MC-objects. The difference between the current image s_{k+1} and synthesized prediction image s^* is estimated for detecting the model failures. The model failures obtained need not be coded with rigid 3-D objects. It is sufficient to code in rigid 2-D objects. Only shape and color parameters are encoded for MF-objects.

Chapter 4

PARAMETER ESTIMATION AND CODING

On completing the initialization with mesh triangles, the shape and motion parameters for the objects are estimated for every successive frame as explained below.

4.1 Estimation of Shape Parameters

In this section, we consider the shape parameter estimation problem of a typical head and shoulder image used in videotelephony. The 3D shape of an object can be determined from 2D silhouette of that object by a distance transformation as dicussed in Sec 3.3. We can simplify the algorithm of detecting the object silhouette for videotelephone sequence using the hypothesis 1 given in

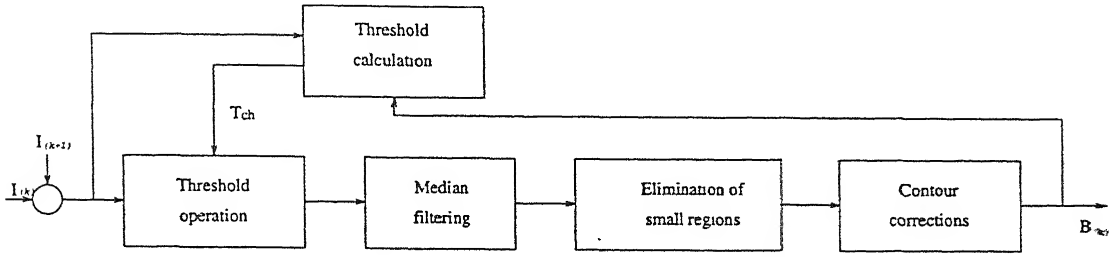

Figure 4.1: Motion segmentation through temporal change detection.

Sec 2.1.

4.1.1 Detection of silhouette of the object

All moving parts of an image are defined as objects and the static areas in an image are referred to as the background. The moving object area can be detected from a temporal image sequence through temporal change detection [5].

Two subsequent frames s_k and s_{k+1} from the image sequence are substracted and a threshold that is adaptive to background image noise is applied. Areas with grey level difference above the threshold are assumed to contain changed areas. In the post processing step, small regions and holes are eliminated by applying normalization. The detected area contains a superimposition of the object silhouette from frame s_k and s_{k+1}, because the object motion from frame k to $k+1$ uncovers parts of the background in s_{k+1}. The area of uncovered background can be detected once the object motion parameters are applied. The detected object silhouette mask is labeled B_k, where its border is the object silhouette. The procedure requires the object to have a nonuniform texture and to move between two images. It may require more than two images to reliably segement the object.

4.2 Estimation of Motion Parameters

To estimate motion of an object, a set of observation points are taken from the object. Each point O^i is described by the position $\mathrm{P}_i = (X_i, Y_i, Z_i)$ on the surface of the object. The position of the observation point in the next frame is denoted as $\mathrm{P}_i' = (X_i', Y_i', Z_i')$. The observation points are taken from the center of the triangles for simplification purposes. 2-D motion of the observation points can be estimated by block matching method. The advantage of using block matching is when implemented in hardware the time required to estimate the 2-D motion will be reduced. The 2-D motion of observation point can be extended to 3-D motion by assuming the object exhibits the rigid 3-D motion. In rigid motion the distance from the point on the surface to the center of the object will not change. Here we can eliminate some observation points considering the position of the points. If the point is on the boundary or very near to the boundary, we can eliminate that as it will give erronous result.

The motion and the depth information for shape updation of the object can be calculated by the following algorithm [1] Each iteration of the algorithm is composed of two steps: 1) determination of motion parameters given the depth estimates from the previous iteration and 2) updation of estimates using the new motion parameters. Let a point (X_i, Y_i, Z_i) at time t_k move to (X_i', Y_i', Z_i') at a time t_{k+1}. It is well known that (X_i, Y_i, Z_i) and (X_i', Y_i', Z_i') can be related, under rigid motion assumption, by

$$\begin{bmatrix} X_i' \\ Y_i' \\ Z_i' \end{bmatrix} = R \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} + T \tag{4.1}$$

where T is the translation vector and R is the rotation matrix. Here rotation and translation are independent of the camera position as the rotation of the object is around its axis. This rotation matrix can be linearized for small angles of R_x, R_y, R_z as

$$[R_C'] = \begin{bmatrix} 1 & -R_z & R_y \\ R_z & 1 & -R_x \\ -R_y & R_x & 1 \end{bmatrix} \tag{4.2}$$

where R_x, R_y and R_z are the rotational angles around x, y and z axes, respectively.

If we take the orthographic projection of Eqn 4.1, we get

$$\begin{bmatrix} x_i' \\ y_i' \end{bmatrix} = \begin{bmatrix} 1 & R_z & -R_y \\ -R_z & 1 & R_x \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ Z_i \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix} \tag{4.3}$$

where (x_i, y_i) and (x_i, y_i) denote the orthographic projections of (X_i, Y_i, Z_i) and (X_i', Y_i', Z_i') to the image plane, respectively.

In (4.3), there are five unknown global motion parameters R_x, R_y, R_z, T_x and T_y and an unknown depth parameter Z_i for a given point (x_i, y_i) and the corresponding (x_i', y_i'). The equation(4.3) has a bilinear nature since Z_i multiplies the motion paramerters. It is thus proposed to solve for the unknowns in two steps:

1. Given atleast three corresponding coordinate pairs (x_i, y_i) and (x_i', y_i') and their depth parameters Z_i, $i = 1, \cdots, N, N \geq 3$, we can rearrange (4.3) in $2N$ equations with five unknowns:

$$\begin{bmatrix} x_i' - x_i \\ y_i' - y_i \end{bmatrix} = \begin{bmatrix} 0 & -Z_i & y_i & 1 & 0 \\ Z_i & 0 & -x_i & 0 & 1 \end{bmatrix} \begin{bmatrix} R_x \\ R_y \\ R_z \\ T_x \\ T_y \end{bmatrix} \tag{4.4}$$

Hence, the motion parameters can be solved from (4.4) using the least squares method. The initial depth estimates are obtained from the scaled wireframe model.

2. Once the motion parameters are found, we can estimatae the new Z_i, values using

$$\begin{bmatrix} x_i' - x_i - R_z y_i - T_x \\ y_i' - y_i + R_z x_i - T_y \end{bmatrix} = \begin{bmatrix} -R_y \\ R_x \end{bmatrix} [Z_i] \qquad (4.5)$$

which is again obtained from (4.3). Here, we have one equation pair per given point correspondence, which can be solved for Z_i in the least square sense.

The procedure consists of repeating steps 1 and 2 until the estimates nolonger change from iteration to iteration.

Due to linearization, motion parameters have to be estimated iteratively for all objects. After every iteration, the model object is moved accordingly. After this a new set of motion equations is established giving new parameter updates. Since motion parameter updates are very small, the introduced linearization do not harm the motion estimation.

4.3 Detection of MF-Objects

The estimated shape and motion are applied to the image s_k and the synthesized prediction image s_k^* is obtained. The next step is to find out the model failures that cannot be described by previously transmitted color parameter. The difference image between synthesized prediction image s_k^* and the current image s_{k+1} is evaluated by binarizing it using an adaptive threshold T_e so that the error varience of the areas which are not declared as synthesis errors is below a given allowed noise level N_e.

The resulting error mask contains both model failures and geometrical distortions. These geometrical distortions can be eliminated by applying 5×5 median filtering. After this process we get small regions, they are named as model failures or MF-object. The total area occupied by model failures is generally less than 4% of the whole image area.

Events in the real world which cannot be modelled by the source model will contribute to synthesis errors. Using the source model of 'moving rigid 3D objects', it is not possible to model changing human facial expressions or specular highlights. Especially facial expressions are subjectively important. In order to be of subjective importance, it is assumed that an erroneous image region has to be larger than 5% of the image area (Fig 5.10). Model failures are those image areas where the model image s_k^* is subjectively wrong (Fig 5.13). Each area of model failures is modelled by MF-2D rigid object defined by color and 2D shape parameters.

4.4 Parameter Encoding

The task of parameter coding is to efficiently code the parameter sets defininng motion, shape and color respectively provided by image analysis to synthesize faithful reproduction of the original image. For MC-objects it is sufficient to code motion and shape parameters. For MF-objects there is no motion and hence it is sufficient for MF-objects to code only shape and color parameters.

4.4.1 Coding of motion parameters

The translation vector is represented in terms of pels and rotation vectors in terms of degrees. These motion parameters are PCM coded by quantizing each

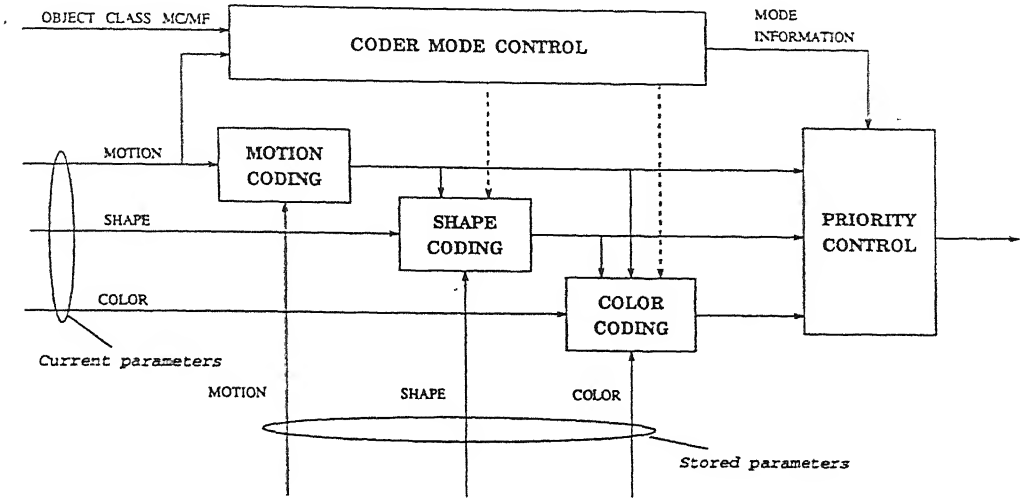

Figure 4.2: Block diagram of parameter coding.

component with 8 bits within an interval of ± 10 pels and ± 10 degrees for translation and rotation vectors respectively.

4.4.2 Coding of shape parameter

The shape can be represented by either polygon or spline. Spline method is computationally intensive than the polygonal approach. But spline method gives a smoothed curve for natural boundaries.

In polygonal approach, first a polygon is generated with four points. The maximum error distance d_{max} from polygon to the actual curve is calculated. If this distance is greater than the allowed quality measure d^*_{max} a vertex for polygon is introduced on the curve at that point and new maximum error distance d_{max} is calculated. This process is continued until we get the polygonal approximation of the required quality. The quality measure d^*_{max} for MC-object is kept 1.4 pels and for MF-object is 2.1 pels.

The shape is represented by the polygen points. The coordinate of the polygens are coded and transmitted. The bitrate for shape parameter depends on the number of polygen points. Overall, we can expect the shape parameter of an object encoding will require about 70 bits for each frame.

4.4.3 Coding of color parameters

Conventional DCT is not suitable for the coding of color parameters of arbitarily shaped regions. New algorithms have been developed for this application. The special type of DCT for arbitarily shaped regions developed by Gilge [8] can be used. The resulting DCT coefficints are quantized with a linear quantizer of signal dependent size. Coding of color parameters has not been attempted in this thesis.

Chapter 5

RESULTS AND DISCUSSIONS

5.1 Results

The OBASC method has been implemented and tested with 'Claire' image sequence. The size of the image sequence taken into consideration is 180×144. The number of frames tested successfully is 3.

First silhouette of the 'Claire' image (fig 5.2) is obtained using the change detection algorithm of Sec 3.3. In the Claire image sequence considered here, only moving part is the head. The shoulder is fixed. This algorithm detected the head as the moving object. The background noise level is kept constant at 7/255. The silhouette is approximated by polygon as shown in fig 5.4. The quality measure d^*_{max} is taken to be 1.4 pels. The contours are drawn on the surface of the model objects initializing the depth with distance transformation as explained in Sec 3.3. This is shown in fig 5.5. The mesh of traiangles are generated from these contour curves as shown in fig 5.6. The Delauny triangulation method is used to generate the triangles. The number of control points as kept constant. It can vary with the size of the silhouette. The texture information of these triangular patches are stored in a buffer. The model scene is projected in the model camera. The initialization part is over at this stage.

3-D motion parameters are estimated as discussed in Sec 4.2. The motion is compensated for the moving parts of the image. Now the silhouette is corrected as discussed in Sec 3.3. Scaled difference between the current image s_{k+1} and prediction synthesis image s^* is obtained (fig 5.7). Applying the adaptive threshold for binarizing the image, the erronous image mask is obtained and is observed that it is not occupying more than 6% of the image area. This is shown in fig 5.8. The 5 × 5 median filtering is applied to the binarized image and geometrical distortion are eliminated. The MF-objects are determined and shown in fig 5.10. It is observed that the area occupied by MF-objects is less than 5% of the image area. The goemetrical distortion are seen in fig 5.9. It is observed that the geometrical distortion occur only near the corners of the image and does not affect the subjective quality of the image.

The image is reconstructed with 1 MC-object and 3 MF-objects. It is shown in fig 5.12 The shape and motion parameters for MC-objects and shape and color parameters for MF-objects are used for the reconstruction of the final image. The actual frame and the decoded frames are shown in fig 5.1b,fig 5.12 respectively.

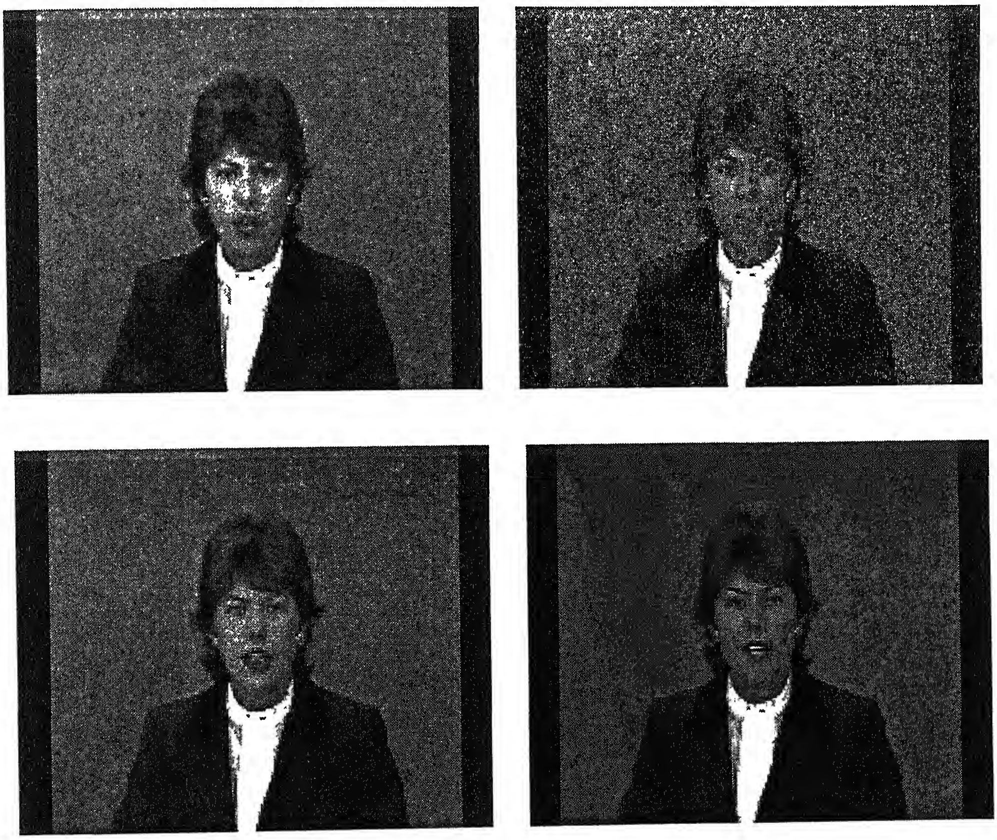

Figure 5.1: Input image sequence.

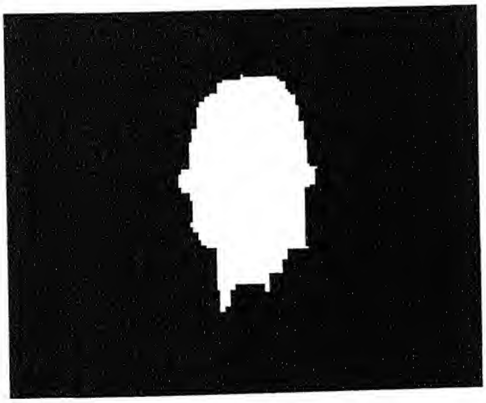

Figure 5.2: Silhouette of the detected object.

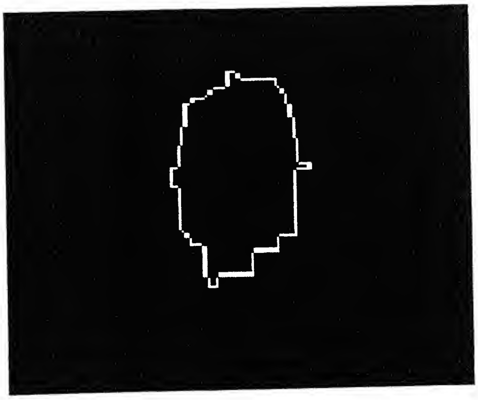

Figure 5.3: Boundary of the detected object.

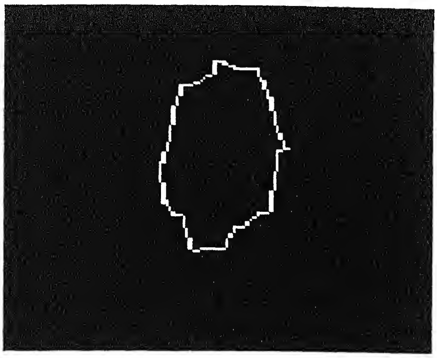

Figure 5.4: Polygon approximation of the object boundry.

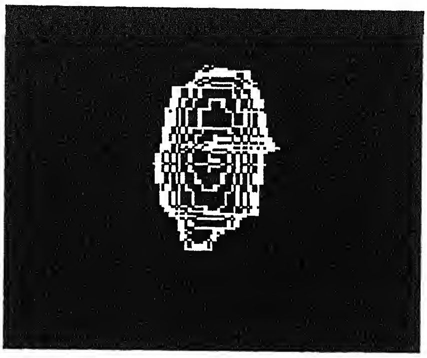

Figure 5.5: Contours drawn on the surface of the object.

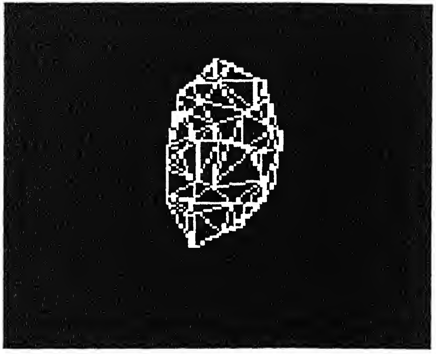

Figure 5.6: Mesh of triangles generated on the object surface.

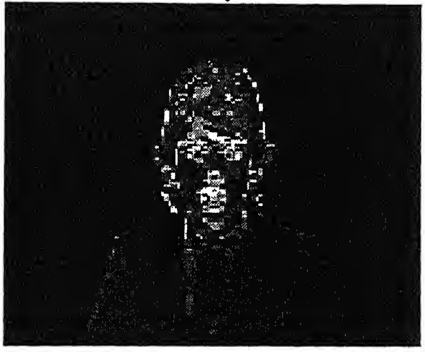

Figure 5.7: Scaled difference between s_{k+1} and s^*.

Figure 5.8: Binarized difference between s_{k+1} and s^*.

Figure 5.9: Geometrical distortion in the s^*.

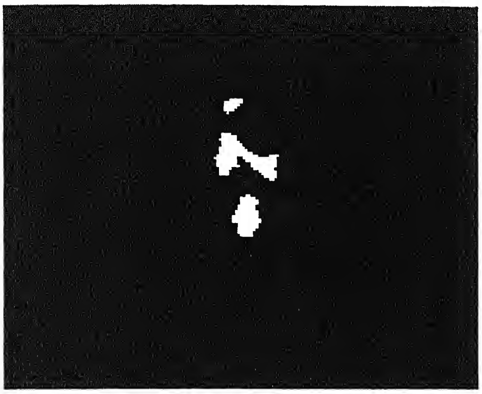

Figure 5.10: Model failure objects.

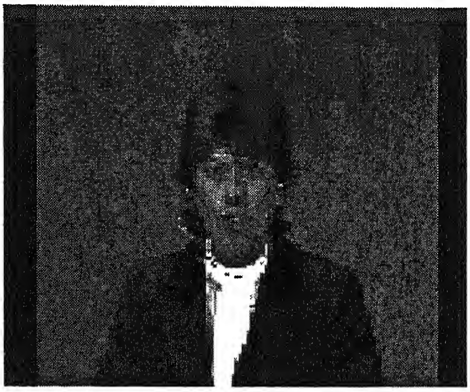

Figure 5.11: Model object projected on the model world.

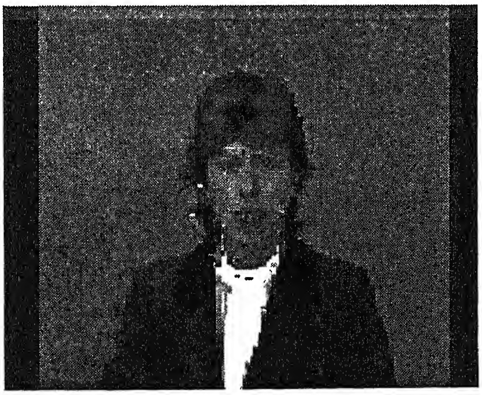

Figure 5.12: Reconstructed 2nd frame of the image sequence.

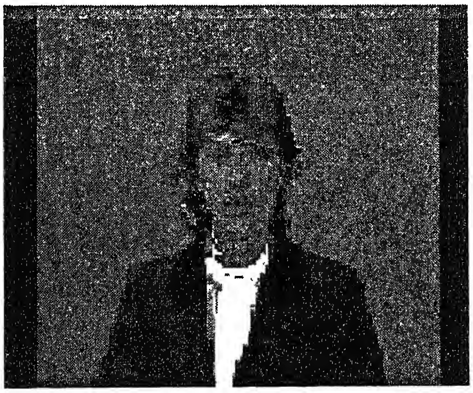

Figure 5.13: Reconstructed 4th frame of the image sequence.

5.2 Discussion

The image sequence is coded with OBASC algorithm. The decoded image is free of any blocking or mosquito effects. However it is observed in the reconstructed image shown in fig 5.13 that distortion do appear near the contours of the image. Although the exact cause of the error has not been identified, it is felt that the distortion may be due to improper depth initialization.

The motion is estimated from the current image frame and the synthesized previous frame. When distortion are present in the synthesized image, it will affect the performance of the coder in parameter estimation. We observe that the although the distortion will not affect the shape parameters, it affects the motion parameters and MF-objects.

5.3 Conclusion

The present work described in this thesis is based on the method suggested by Ostermann [2]. However, the present implementation differs from that given in [2] in the following aspects:

1. The texture mapping of MC-objects

2. Motion and depth estimation of the MC-objects.

The texture mapping here uses the color information that is stored in buffer separately for different triangular patches. The triangular patch vertices are transformed according to the motion and texture is applied accordingly.

The 2-D motion for observation points are estimated. Some of the points are eliminated depending on the position and motion of the points to get robust estimation. Global 3-D motion is estimated from these motion by an averaging method that reduces the mean estimation error. The estimation error has strong correlation with depth and this depth is estimated. This depth will be used to estimate the shape parameters.

This method is implemented and tested with a standard 'Claire' image sequence. The area of MF-objects turns out to be 5%. Hence the data to represent the image sequence is very low while preserving the quality of the image. It does not have the usual blocking and mosquito effects.

5.4 Scope for Future Work

This work can be extended by including the following:

1. 'DCT for arbitary shape' can be used to encode the color parameters of MF-object. The performance of the algorithm in terms of bitrate can improved. The waveform coding method can also be applied for encoding the color parameters of MF-objects.

2. The source model taken here is rigid 3-D object. Flexible 3-D objects can be chosen instead, to increase the quality of the reconstructed image. For this source model, one more parameter for 'flexible shape' has to be added. But we can expect this will reduce the size of MF-object area and hence overall bitrate will be compensated. Flexible shape parameters can be estimated with residual error along with the motion parameters.

3. The shape parameter is approximated by polygons. To get smoothed boundaries of the object, splines can be used. The data required will be nearly the same. The quality of the image can be improved at the expense of increased complexity.

4. It is not sufficient to use a single facial image alone. This is because it is impossible for texture image to have the mouth closed and open simultaneously. To overcome this problem we can employ several texture images.

Bibliography

[1] A. Murat Tekalp,Gözde Bozdaği and Levent Onural. An improvement to mbasic algorithm for 3-d motion and depth estimation. *IEEE Transactions on Image Processing*, 3(5):711–716, September 1994.

[2] Astrid Lundmark,Haibo Li and Robert Forchheimer. Image sequence coding at very low bitrates: A review. *IEEE Transactions on Image Processing*, 3(5):589–608, September 1994.

[3] Pertti Roivainen,Haibo Li and Robert Forchheimer. 3-d motion estimation in model-based facial image coding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):545–555, June 1993.

[4] Peter Pirsch Hans,George Musmann and Hans-Joachim Grallert. Advances in picture coding. *Proceedings of the IEEE*, 73(4):523–548, April 1985.

[5] Michael Hötter and Robert Thoma. Image segmentation based on object oriented mapping parameter estimation. *Signal Processing*, 15(3):315–334, October 1988.

[6] Reinhard Koch. Dynamic 3-d scene analysis through synthesis feedback control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):556–568, June 1993.

[7] A. Murat Tekalp,Gözde Bozdaği and Levent Onural, "An improvement to mbasic algorithm for 3-d motion and depth estimation," *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 711–716, September 1994.

[8] M.Gilge, T.Engelhardt, and R.Mehlan, "Coding of arbitrarily shaped image segments based on a generalized orthogonal transform," *Signal Processing: Image communication*, vol. 1, no. 2, pp. 153–180, October 1989.

[9] Jörn Ostermann, "An analysis-synthesis coder based on moving flexible 3-d objects," in *Proc. Picture Coding Symp.*, March 1993.

[10] Jörn Ostermann, "Object-based analysis-synthesis coding (obasc) based on the source model of moving flexible 3-d objects," *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 705–711, September 1994.

[11] Peter Pirsch Hans,George Musmann and Hans-Joachim Grallert, "Advances in picture coding," *Proceedings of the IEEE*, vol. 73, no. 4, pp. 523–548, April 1985.

[12] Frederic I. Parke, "Parameterized models for facial animation," *IEEE Transactions on Computer Graphics*, pp. 61–68, November 1982.

[13] Pertti Roivainen,Haibo Li and Robert Forchheimer, "3-d motion estimation in model-based facial image coding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 6, pp. 545–555, June 1993.

Appendix A

The use of generating function for computing z-dimension allows, one to generate a model object from an arbitrarily shaped silhouette. For the head and shoulder images, ellipse is a better approximation for depth information. For videophone test sequences, subjectively good results [2] are achieved with the ratio of object width to object depth set to 1.5(Fig A.1).

The resulting 3-D shape is approximated by contour curves such that the distance between two contour curves along the surface of the object is constant.

The distance between two points (acos θ_1,bsin θ_1) and (acos θ_2,bsin θ_2) on the surface of the ellipse is given by the equation

$$s = \int_{\theta_1}^{\theta_2} \sqrt{\left(\frac{dx}{d\theta}\right)^2 + \left(\frac{dy}{d\theta}\right)^2} d\theta$$

The contour curve points are calculated by integrating the above equation numerically.

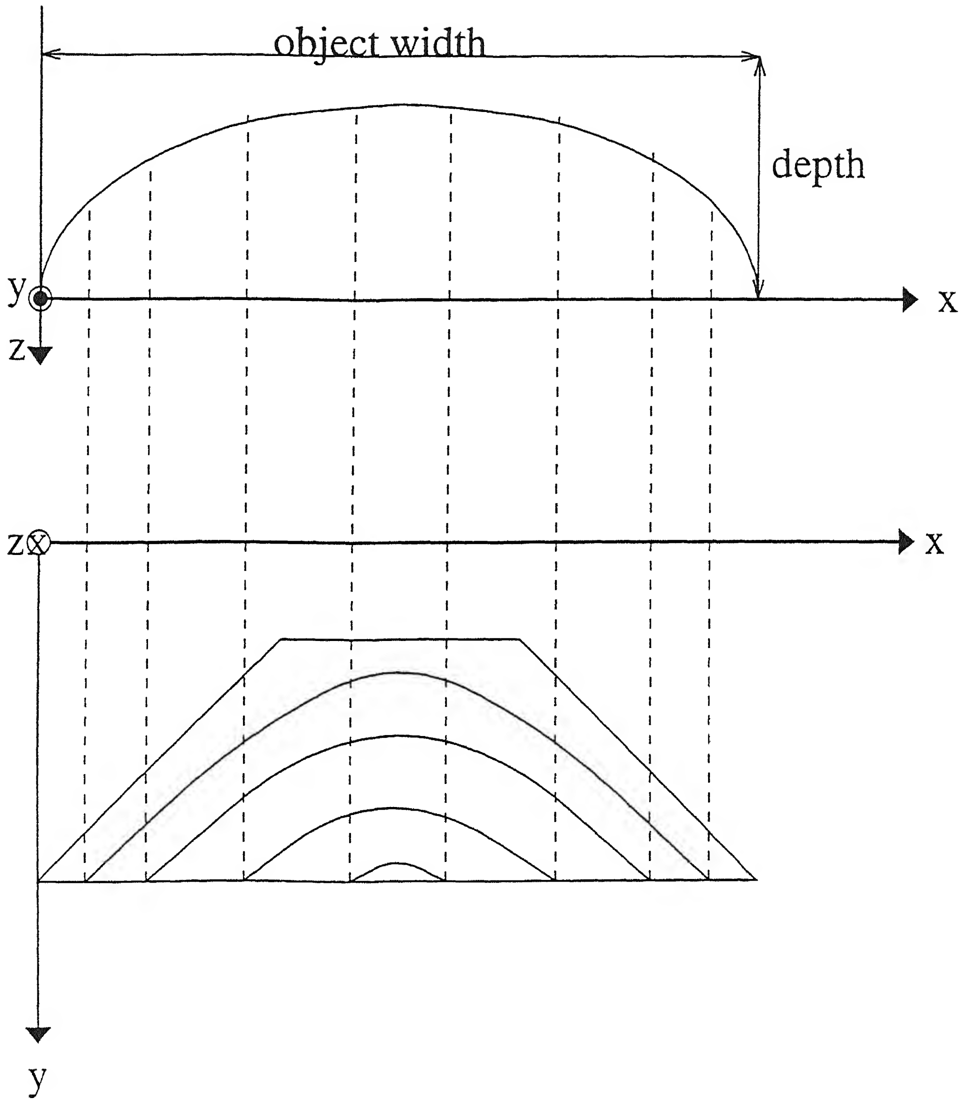

Figure A.1: Function giving z-coordinates of the 3-D shpae and contour lines.